

```
import numpy as np
import pandas as pd
df = pd.read_csv('mail_data.csv') # Load CSV file
print(df.head()) # 1. Show first 5 rows
```

```
↵ Category      Message
0      ham  Go until jurong point, crazy.. Available only ...
1      ham                Ok lar... Joking wif u oni...
2     spam  Free entry in 2 a wkly comp to win FA Cup fina...
3      ham  U dun say so early hor... U c already then say...
4      ham  Nah I don't think he goes to usf, he lives aro...
```

```
# 2. Find total rows and columns
print(df.shape)
```

```
↵ (5572, 2)
```

```
#3. Show column names
print(df.columns.tolist())
```

```
↵ ['Category', 'Message']
```

```
#4. Check missing/null values
print(df.isnull().sum())
```

```
↵ Category      0
   Message      0
   dtype: int64
```


```
#5 Find number of spam and ham messages
print(df['Category'].value_counts())
```

```
↵ Category
   ham      4825
   spam      747
   Name: count, dtype: int64
```


```
#6 Find % of spam messages
print((df['Category'].value_counts()['spam'] / df.shape[0]) * 100)
```

```
↵ 13.406317300789663
```

```
#7 Find % of ham messages
print((df['Category'].value_counts()['ham'] / df.shape[0]) * 100)
```

 86.59368269921033

```
#8 Average length of a message
average_length = df['Message'].apply(len).mean()
print(average_length)
```

 80.36898779612348


```
# 9. Maximum message length
max_length = df['Message'].apply(len).max()
print(max_length)
```

 910


```
# 10. Minimum message length
min_length = df['Message'].apply(len).min()
print(min_length)
```

 2


```
#11 total number of messages?"
total_messages = len(df)
print(total_messages)
```

 5572

```
#12 Messages longer than 100 characters
messages_above_100 = (df['Message'].apply(len) > 100).sum()
print(messages_above_100)
```

 1761

```
# 13. Shortest message text
shortest_text = df.loc[df['Message'].apply(len).idxmin()]
print(shortest_text)
```

 Category ham
Message Ok
Name: 1925, dtype: object

```
# 14. Number of duplicate messages
duplicate_messages = df.duplicated(subset=['Message']).sum()
print(duplicate_messages)
```

 415

```
#15 # 16. Spam messages containing the word 'win'
spam_with_win = df[(df['Category'] == 'spam') & (df['Message'].str.contains('win', case=False))].shape[0]
print(spam_with_win)
```

 100

```
# 16. Show the shape of the dataset
print("\nShape of the dataset:")
print(df.shape)
```



```
Shape of the dataset:
(5572, 2)
```

```
# 17. Check if any message is completely in uppercase.
print("Are there any fully uppercase messages?")
uppercase_messages = (df['Message'].str.isupper()).sum()
print(uppercase_messages)
```



```
Are there any fully uppercase messages?
97
```

```
# 18. Check if any message is completely in uppercase.
uppercase_messages = (df['Message'].str.isupper()).sum()
print(uppercase_messages)
```




```
97
```

```
#19. Find the top 10 most common words in spam messages.
spam_words = ' '.join(df[df['Category'] == 'spam']['Message']).lower().split()
top_spam_words = pd.Series(spam_words).value_counts().head(10)
print(top_spam_words)
```



```
to      682
a       375
call    339
your    263
you     252
for     202
the     201
or      188
free    180
2       169
Name: count, dtype: int64
```

```
# 20. How many messages contain numbers (0-9)?  
messages_with_numbers = df['Message'].str.contains(r'\d').sum()  
print(messages_with_numbers)
```

 1460